

EasyZoom: Zoom-in-Context Views for Exploring Large Collections of Images

Jiajian Chen

Yan Xu

Greg Turk

John Stasko

School of Interactive Computing
College of Computing
Georgia Institute of Technology
85 5th St, Atlanta, GA
{johnchen, yan.xu, turk, stasko}@cc.gatech.edu

ABSTRACT

Image browsing and searching are some of the most common tasks in daily computer use. Zooming techniques are important for image searching and browsing in a large collection of thumbnail images in a single screen. In this paper we investigate the design and usability of different zoom-in-context views for image browsing and searching. We present two new zoom-in-context views, sliding and expanding views, that can help users explore a large collection of images more efficiently and enjoyably. In the sliding view the zoomed image moves its neighbors away vertically and horizontally. In the expanding view, the nearby images are pushed away in all directions, and this method uses a Voronoi diagram to compute the positions of the neighbors. We also present the results of a user study that compared the usability of the two zoom-in-context views and an overlapping, non-context zoom in the tasks of searching to match an image or a text description, and the task of brochure making. Although the task completion times were not significantly different, users expressed a preference for the zoom-in-context methods over the standard non-context zoom for text-matching image search and for image browsing tasks.

Author Keywords

Image search, image browsing, image collections, focus+context.

ACM Classification Keywords

H5.2 Information interfaces and presentation: User Interfaces. - Graphical user interfaces.

General Terms

Algorithms, Design, Experimentation.

INTRODUCTION

Large image collections are becoming common both in

research and for personal use, due to several converging trends. The recent availability of digital cameras, both consumer grade and at the high end, have been the most important contributing factor in the creation of these large image databases. In addition, the low cost of mass storage and the prevalence of image hosts on the web, such as Google Picasa and Flickr, have also contributed to this trend. It is now common for casual photographers to amass photo collections that contain tens of thousands of images. A key technique for browsing and searching such a large photo collection is the combination of thumbnail images and image zoom. Our work is aimed at developing new methods for performing image zoom and for analyzing the effectiveness of these new techniques.

We began our research from the strategy of *focus in context*, taken from the HCI and information visualization communities. The practice of focus in context is to give a user the ability to closely examine a visual representation of details in their data (*focus*), while keeping this information visually near related information (*context*). For the application of searching and browsing in image collections, our goal is to provide methods of enlarging an image, while keeping neighboring thumbnail images in nearly the same relative position to the zoomed image as before. Our hypothesis is that users will find such zoom-in-context methods to provide a better photo browsing experience.

There are many possible ways in which to embody the idea of zoom-in-context in an image browsing system. We began our work by designing a number of techniques to enlarge a user-selected image so that nearby images stay visible. We then went through several design iterations for these new methods to refine the interaction. After refining several zoom-in-context methods, we selected the two methods that we found the most promising, and ran a user study to gauge their effectiveness. In the remainder of this paper we review the related work, we describe the designs and implementations of several zoom-in-context methods, and we present the user study of the two best methods.

RELATED WORK

There are two areas of research that are closely related to our work, the area of user interfaces for photo browsing and

GVU Tech Report
GIT-GVU-12-02
February 2012

the study of focus-in-context techniques. We will review work from each of these areas in turn.

User Interfaces for Photo Browsing

Many researchers have studied user interfaces for browsing collections of photographs on a computer screen, and we cover only some of them here.

Some researchers build image browsing systems by showing photos in groups that are clustered by time-stamp or events. PhotoTOC is a system that automatically clusters a user's personal photo collection according to the photo time-stamps [12]. When browsing, representative images can be used to act as stand-ins for clusters. A study indicated that users preferred this form of indexing over a hierarchical browser and a standard scrollable list of images. Graham and co-workers found time-stamps to be useful in organizing images into clusters in personal photo collections [5] as well. Their Calendar Browser system is organized in terms of years and months, and it uses representative photos for these clusters. When compared to hierarchical and scrollable browsers, their time-based browser allowed users to perform more rapid searches. [3] also used time stamp as a principle metric to group and visualize photos. Yang and colleagues developed a Semantic Image Browser that uses semantic image analysis to characterize photos as multivariate data objects [21]. The system then uses multi-dimensional scaling and the Value-and-Relation information visualization technique to project the images into a window. The viewer can use the clustering of images for browsing the collection, and the system provides a number of flexible interface techniques for examining the photos or the visualizations.

Time Quilt is a system that combines the benefits of time-stamps and scrollable lists of images into one system [8]. In effect, the system takes a long time-ordered column of photos and folds this into multiple columns to use screen space more effectively. With a goal of locating a particular image, their system was compared against other methods, and searching time was reduced in most cases. PhotoArcs [1] visualizes photos by creating photo-narratives. Torres et al. use spiral and concentric ring layouts to control the presentation of images [20]. Photos near the center are more in focus and thus larger while images toward the periphery are displayed smaller. Kustanowitz and Shneiderman explore a two-level radial display for browsing photos [10]. The center region is occupied by a main focus image that is largest. Smaller clusters of images are then laid-out radially around the main image.

Another approach to photo browsing typically uses layout techniques such as grids of thumbnails, but augments the layouts with query and interaction capabilities for controlling which images are presented [9]. The PhotoFinder system provides a variety of visual Boolean query interfaces as well as dynamic query and query preview features to select the images being shown. The

system also allows photos to be placed onto a scatter plot where the dimensions can describe characteristics of the pictures. Smith and colleagues are exploring faceted search capabilities to assist people browsing for various kinds of data, where images may be one important example [19]. The FacetMap system displays search results in a treemap-like information visualization. User selections then filter the result set and the display presents animated view transitions in order to provide a fluid, flexible interactive exploration capability.

Other research in the area has stressed the study of people's current habits and preferences in organizing and browsing photos to better inform the design of future systems. Rodden and co-workers examined whether organizing by visual or caption similarity was helpful in photo browsing [15]. They found that users found arrangements by caption similarity (keywords) and by visual similarity were both useful, with caption similarity being slightly preferred. Most users agreed that it was useful to have more than one kind of view, and this is a result that will inform the design of our own image browsing systems. The researchers noted that in some situations, users scanned through the full set of images on the screen in search of a suitable image. Rodden and Wood examined how people used a digital photo management system over the course of six months [16]. They found that much of the preference for using a digital system over a non-digital storage system was due to the browsing features that a digital system enables.

Perhaps the research most closely related to our own is Bederson's photo browsing system PhotoMesa [2]. PhotoMesa is designed to create 2D layouts of photo collections that group photos based on the user's own clustering of the photos. The system attempts to remain faithful to these groups while displaying many thumbnail images in a manner that also makes full use of the 2D space available on the screen. The user may click within a group, and the system zooms up just those images in this group. When the cursor dwells over a particular image for a sufficiently long time, the image is zoomed to 200 pixels in size. This zoomed image overlaps and obscures the surrounding images. When the cursor is moved again, the image reduces to thumbnail size.

Focus-in-Context Interactive Techniques

Numerous researchers have explored methods of interaction that magnify one portion of a data collection while retaining a reduced-size context that surrounds the enlarged region. We do not present a comprehensive survey of such techniques, but we describe selected papers based on this principle. Perhaps the earliest work to recognize the general principle of focus-in-context is the fisheye view by [4]. Robertson et al. followed this principal for their work in visualizing hierarchical information with cone trees [13] and in their visualization work for wide 2D layouts using the perspective wall [11]. Fisheye views were also used by Sarkar and Brown for viewing graphs [17]. Robertson and

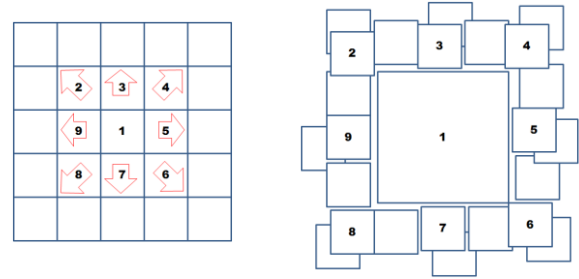
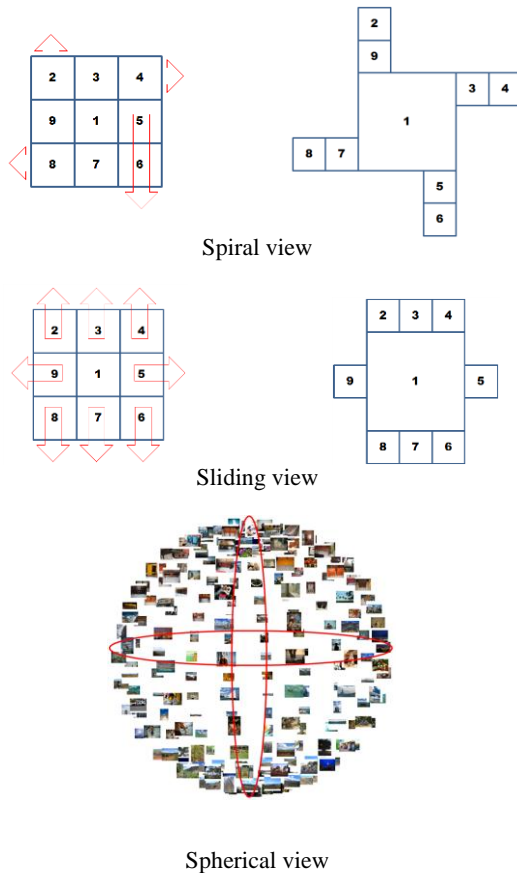
Mackinlay presented the document lens as a method of focus for documents with 2D layouts that are both wide and high [14].

After reviewing the literature in both areas, we found many image browsing systems use only the common non zoom-in-context view, such as PhotoMesa and ACDSee. Some systems show image collections in a 3D manner and provide a detail view for each single image instead of using zoom-in-context technique. However, we found there is little existing work that's focused on designing zoom-in-context views for image thumbnails that are tiled as 2D array on the screen, or evaluating different zoom-in-context views in an image browser. Our goal is to design zoom-in-context views for image browsers and evaluate their usability.

DESIGN OF ZOOM-IN-CONTEXT VIEWS

First Iteration of Design

In the first iteration of design of zoom-in-context views, we tried several possibilities, including a sliding view, spiral view, spiral view, spherical view, expanding view and their variations. Figure 1 illustrates each of the views.



Expanding view

Figure 1. Several zoom-in-context views we prototyped in the first iteration of design

In the spiral view the neighbors of the zoomed image slide away in four directions and the arrangement results in a spiral after zooming. In the sliding view the zoomed image's neighbors move away horizontally and vertically along rows and columns. In the spherical view thumbnails are located on the surface of a 3D sphere in a perspective view. The user can rotate the sphere by dragging and moving the mouse and the thumbnails always face toward the viewer. In this way a thumbnail image is automatically zoomed in when it moves closer to the viewer. In the expanding view the zoomed image's neighbors move away in all directions and the arrangement after zooming still exhibits a loose grid pattern.

After a first round of prototyping and testing, we decided to discard the spiral view and the spherical view due to disadvantages in each of these views. The spiral view's sliding pattern is interesting, but it introduces too much visual distraction especially when a user continuously zooms in to multiple images that are close to one another. We also felt that this method changes the spatial relationship of the images too much. We felt that the spherical view does not enlarge the center image enough even when this image is already the closest one to the user. That is, using perspective alone is not enough to provide a significant size difference between the center and peripheral images. The spherical view shows images in 3D, which also introduces confusion for user interaction, especially when users rotate the sphere. We also tried a fisheye-like view, in which the center image is enlarged and its neighbors are shrunk based on their degree of interest (DOI), or distance to the center image. This view shrinks the images with low DOI too much so we also discarded it.

After the first iteration of design, prototyping and testing, we selected the sliding view and expanding view for further study. Both these methods zoom the center image in an aesthetically-pleasing manner and they preserve the neighbor relationships well.

Sliding View: Design and Algorithm

In the sliding view the zoomed image's neighbors move away horizontally and vertically. There are several variations of the sliding view in Figure 2.

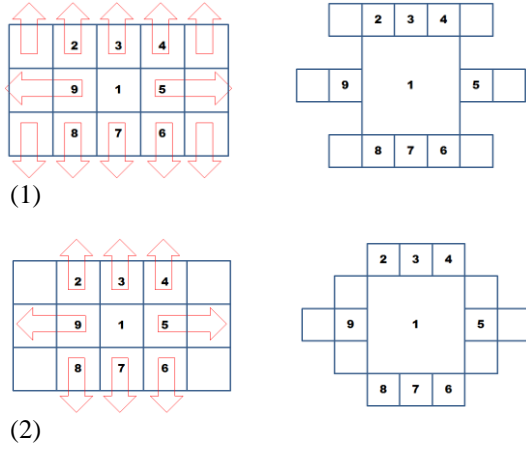


Figure 2. Variations of the sliding view

We can obtain another two variations by flipping the two cases above by 90 degrees. From early prototypes and tests, we decided to choose (2) in Figure 2 out of these four variations, because it moves the smallest number of neighbors thus minimizing visual distraction in the zooming animation. Also it uses screen space better, and it looks more aesthetically-pleasing.

Assume the images on a single screen are indexed from bottom to top and from left to right and each image located at (x,y) has a row index i and a column index j . In the sliding animation the center image is enlarged, and the moved images are only those in the three columns containing or near the zoomed image, and the images in the image's row.

Pseudo-code that describes the sliding animation is as follows.

```

For each step, do the following
Increase the size of the center image at  $(i,j)$ 
For images whose row== $i+1$  &&  $(col \geq j-1 || col \leq j+1)$ 
    update  $y += \text{delta}$ ,
For images whose row== $i-1$  &&  $(col \geq j-1 || col \leq j+1)$ 
    update  $y -= \text{delta}$ ,
For images whose row== $i$  &&  $col \leq j-1$  update  $x -= \text{delta}$ ,
For images whose row== $i$  &&  $col \geq j+1$  update  $x += \text{delta}$ ,
Draw all images to the screen

```

When the user selects an image to zoom while another image is already zoomed (e.g., the user clicks image 1 first, and then clicks its neighbor image 5), the previously zoomed image must shrink back to its original size and the currently zoomed image must enlarge. We overlap in time

the shrinking and enlarging of these two images, as well as overlapping the motions of their neighbors. This means that the newly selected image begins to enlarge immediately, rather than waiting until a previously zoomed image has been reduced to thumbnail size. We do this with the use of linear interpolation to compute the transition that blends the two zooming animations. We first record the position and size of each thumbnail image i in the first sequence of zooming animation as follows.

$$f_i(0), f_i(1), f_i(2), \dots, f_i(MAX)$$

$f_i(t)$ is a multi-dimensional vector that contains the position and size of the i -th image on the screen at the t -th step.

$$f_i(t) = [x \ y \ w \ h]$$

MAX is the maximum number of steps in the zooming animation. Second we compute the position of size of each thumbnail image as if there is only one new image to be enlarged. The new position and size of each image in the second sequence of animation is denoted as follows.

$$g_i(0), g_i(1), g_i(2), \dots, g_i(MAX)$$

Finally the position and size of each thumbnail image in the transition between two zooming animations is computed by the following blending function.

$$h_i(t) = \left(1 - \frac{t}{MAX}\right) \times f_i(MAX - t) + \left(\frac{t}{MAX}\right) \times g_i(t)$$

where $0 \leq t \leq MAX$

The figure below illustrates the transition between two zoomed images 1 and 5.



Figure 3. Transition that blends two zooming animations

By doing this we smoothly blend the shrinking and enlarging animation when the user selects new images to zoom. This method can be used in any zoom-in-context view to blend the two sequences of zoom animations. In particular, we also use this method in the implementation of the expanding zoom.

Expanding View: Design and Algorithm

Our design for the expanding view comes from the observation of grid-tiled images on a single screen. In almost all image browsing software, thumbnail images are arranged in a regular 2D array. The aspect ratio of each thumbnail image is typically 4:3 or 16:9, and usually each thumbnail is placed within a square region that is the same size for all thumbnails. This arrangement means that at least 25% (4:3) or 43% (16:9) of the screen is used as padding space that does not contain pixel information from images. Our idea is to make use of this extra space for a zoom-in-context view. In our expanding view, the enlarged image pushes its neighboring images away in all directions, but without any size reduction of the neighbors.

In our search for an implementation technique for the expanding view, we looked for a geometric placement technique that arranges graphical elements in a natural manner based on their size and shape. We found a suitable method for such geometric placement in the *central Voronoi diagram* method [6]. Given a collection of points P in the 2D plane, a *Voronoi cell* for one of these points is the convex region that is the union of all positions that are nearer to this point than to any other point in P . The collection of all such Voronoi cells is called the *Voronoi diagram* for the points P . Hausner discovered that the Voronoi diagram can be used for packing disk-shaped geometric elements in the plane by a simple iterative technique. Each disk is represented by a point, and the Voronoi for the set of initial point positions is created. Then, each point is given a new position based on the centroid of its Voronoi cell. This process is repeated either until the points no longer move or after a fixed number of iterations. Two points that start out close to one another are pushed away because the centroid of their Voronoi cells are in the direction opposite the closest neighbor.

2D Voronoi diagrams can be calculated rapidly using graphics hardware [7]. The idea is to rasterize a set of cones into a depth buffer, one cone per point that is to be placed. Each of the cones is given a distinct color, so that the image that is formed identifies all of the pixels that are nearest to a particular point by color. If we reduce (or increase) the slope of a cone, the size of its projection will also become larger (or smaller), as shown in Figure 4. In this figure, three cones are drawn to the frame buffer, and they are rendered in an orthographic view from top. In (c) the three cones have the same slope so their projection takes same area in (a). In (d) the center blue cone has a smaller slope so its projection area is bigger than those of other two cones in (b).

The cones in Figure 3 are circularly symmetric, which is appropriate for a Euclidean distance metric. Hausner noted, however, that nearly rectangular or square regions can be created using the Manhattan distance, which is done using the appropriate “cones” for this metric, which are in fact four-side pyramids [6]. We use such pyramidal cones for

performing our placement of images for our expanding zoom. Each iteration of the placement method first rasterizes the cones, then re-computes the average position of pixels in each region, and finally updates the Voronoi cell centers. This process converges and generates a stable pattern in which a single large region is encircled by many small regions.

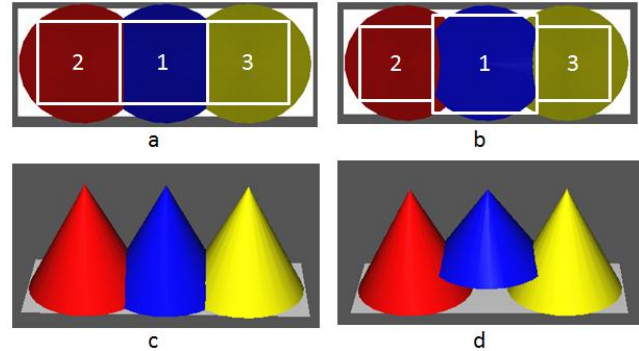


Figure 4. Three cones are rendered in the frame buffer (c, d) in an orthographic view from top (a, b). Note that the projection of a wider cone in (d) is bigger than the projection of other two cones(b). The squares with white boundary in (a, b) are the actual area of the projection of cones, in which we can place images.

If we re-compute the average position of pixels in each region, update the Voronoi region centers, and redraw the cones, this process will converge and generate a stable pattern in which a single large region is encircled by many small regions.

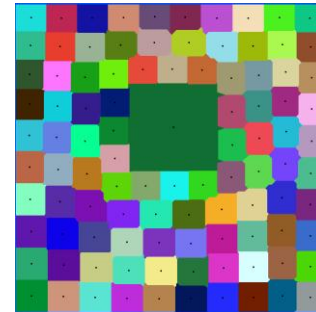


Figure 5. 100 cones in the frame buffer from a top view.

As shown above, initially all cones have the same slope and their projections divide the screen into a 10x10 grid. Then we change the slope of a cone, re-compute the region centers, and redraw cones. The new pattern is stable after 50 iterations. In the new pattern, the large green region is the projection of the cone whose slope is about 1/3 of the slope of other cones.

Given this pattern, we can embed image thumbnails in each region to create the zoom-in-context view. The image embedded in the large green region is the zoomed image, which is

surrounded by its neighbors. Also, the process of updating the Voronoi region centers naturally generates the zooming animation. The procedure of our algorithm for the expanding view is listed below, assuming users select the i -th image to be zoomed.

1. Rasterize cones with the same slope to the frame buffer
2. Repeat until we reach the max steps in the zoom animation
 - 2a. Read back the frame buffer. Compute the average position of all pixels in each Voronoi region and use them as the center position for placing images. If the j -th image is one of the eight neighbor images, add small spring force to it and modify the position of j -th Voronoi center to pull it towards the i -th region.
 - 2b. Draw the enlarged center image and other images at the new Voronoi region centers
 - 2c. Reduce the slope of the i -th cone in each step. Rasterize all cones to the new Voronoi centers again.

Typically after a certain number of steps (e.g., 50 steps) the Voronoi cell centers will converge and the process generates a nice pattern with the enlarged center image encircled by its neighbors. Note that in the step 2a we add a small constant force to pull the eight neighboring images towards the enlarged center image. The purpose of this is to keep the spatial relationship between the center image and its contextual neighbors.

The figure below illustrates the arrangement of images before and after the expanding zoom. The neighborhood relationship is preserved during and after the zoom.

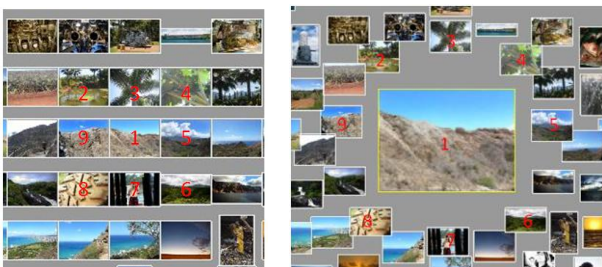


Figure 6. The arrangement of images before and after zooming in the expanding view

The centroidal Voronoi diagram-based algorithm elegantly uses the padding space to push neighbor images away and make room for the zoomed image. After zooming, the thumbnails do not overlap or overlap only a small amount. Another nice aspect of using the Voronoi diagram is that it almost never generates exactly the same zooming pattern for different images, which is quite different from any other

zoom-in-context views. Hence it provides a fluid feeling interaction, compared to the rigid zooming patterns produced by other views.

We also use the same linear function described in the previous section to blend the animation when users continuously select and zoom different images.

Implementation

In the prototype of all these zoom-in-context views, we use OpenGL to render the images and zooming animations. Image thumbnails are uploaded to graphics card memory and stored in texture memory. The program uses orthographic projection of textured rectangles to draw the images. We chose OpenGL instead of Windows GDI because of the hardware graphics acceleration, this method can easily render thousands of images with complex animation (e.g., zooming and moving) smoothly without any flickering.

We need to read back the frame buffer to compute the positions of thumbnails in the expanding view. The frame buffer read back is a slow process, because it blocks the rendering pipeline and copies data from the graphics card to the CPU. Our method is to render the cones and compute the Voronoi region centers at a low resolution frame buffer (e.g., 360×240), normalize the positions to $[0,1]$, and then draw images to a larger screen (e.g., 1680×1050). This approach gives real-time performance on a desktop Windows PC with 2.2GHz Xenon CPU and an nVidia 7950G graphics card.

USER STUDY

Subjects and Image Collections

In order to evaluate the effectiveness of different zoom-in-context methods, we performed a user study of three zoom methods. We recruited 15 people to participate in the study of our image browser with three different zoom styles (overlapping, sliding, and expanding zooms). All the users were computer science graduate students, and they were all proficient at using computers. As images for the study, we collected 360 images for each of 10 cities from the photo sharing web site flickr.com. These photos were from the following cities: New York, Amsterdam, Paris, Venice, Honolulu, Fiji, Stockholm, Beijing, Seoul and Tokyo.

To keep consistency during the study, we showed the 360 images at a 24×15 matrix on the full screen, as shown in Figure 7. The size of the screen was 20 inches and its resolution is 1680×1050 . The image thumbnail size was chosen so that in the searching tasks users needed to zoom in from time to time to find the correct result. The top region of the screen is used to place the images that the user selected.



Figure 7. The screenshot of the program interface in the user study. All images are of Paris, downloaded from flickr.com

The system provides two zooming methods: the left-clicking and dwelling modes. In the left-clicking mode, users must click the image they want to enlarge. In the dwelling mode, any image that the cursor hovers over will automatically be zoomed, without the need for left-clicking. We forced the participants in our study to use the dwelling mode because we wanted to make sure that they were using the zooming features in the system at all times. Moreover, the dwelling mode was preferred by pilot subjects who tried the system before we ran our formal study.

During the study, a participant makes a selection by right-clicking on the image thumbnail. Once selected, the image smoothly flies to the top portion of the screen. Also, participants could right-click the image on the top to remove it from the list. We record the times when the participants right-clicks a given image. Right-clicking (for selection) and zooming an image were the only actions that a participant could perform in the study.

Methodology

The goal of our study was to investigate the usability of three different zoom styles for images (overlapping, sliding, and expanding views) in an image browser. The non zoom-in-context overlapping view is used in many different image browsing systems, such as ACDSee and PhotoMesa, and it acts as a control condition in the study. Our implementation of the overlapping view is in Figure 8.

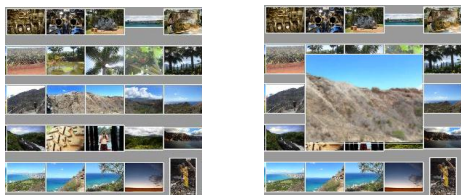


Figure 8. Overlapping view

Participants used our image browser to interact with a large collection of images. We logged the time of all operations.

After the experiment, participants were informally interviewed by the researchers. Each session lasts about one hour.

During the study, participants were sitting in front of a PC with the image browser application running. By moving the mouse and hovering it on a picture, the user could see the zoomed image. In the tasks of search-by-image and search-by-text, we provided the search criteria on a separate screen adjacent to the PC as shown in Figure 9.

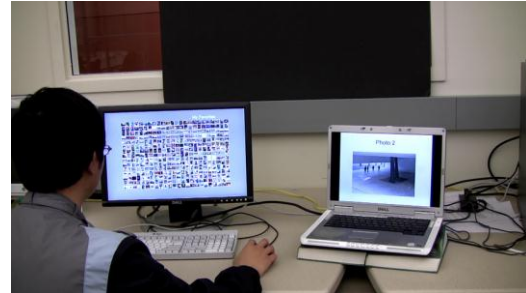


Figure 9. The setting for search-by-image and search-by-text tasks. The target image or text descriptions are shown on the right laptop, while the image set is shown on the left screen.

For this study, each participant performed three separate tasks with the image browser: search-by-image, search-by-text, and brochure making. For *search-by-image*, the participant was presented with a large image on the adjacent screen. The participant was asked to find an exact match to this image within the collection of thumbnails. Once a participant decided that the correct image had been found, they selected it with a right-click on the image. The *search-by-text* task is similar, but a short phrase (rather than a target image) was presented to the user on the adjacent screen.

The *brochure making* task is more elaborate than the two search tasks. For this task, the participants were told that they must select 10 images to be used in a travel brochure for a given city. They were instructed to select images that they think would be appropriate for such a brochure. We adopted this task from the photo browsing research of [16].

The procedure for a complete session was as follows:

- 1) *Tutorial*: Participants were asked to go through a brief tutorial of the image browser. This tutorial contained all of the necessary information about the program features, such as image zooming and selecting a favorite image.
- 2) *Tasks*: Participants were asked to perform each of the three tasks (search-by-image, search-by-text, brochure making) using each of the three zoom techniques (overlapping, sliding, expanding). The task results such as searching time and the index of selected images were logged by the browser.
- 3) *Questionnaire and semi-structured interview*: After each of the tasks, the subject filled out a written questionnaire

consisting of likert scale questions about *efficiency*, *ease of use* and *enjoyment* with 1-7 scales.

After completing all three of the tasks and finishing the written questionnaire, each participant was informally interviewed. During this the interview session we asked the participants to discuss several topics including: Discuss the pros and cons of each view and which view was their favorite; Point out aspects of the browser that they liked, disliked, or found confusing; Describe how this browser compares to other browsing software they had used (e.g., PhotoMesa or ACDSee), or if they had used no other browsers, we asked if they would like to use this browser. The participants were encouraged to describe any changes or new features they would like for the browser.

We used the New York image set for the tutorial at the beginning of the task. The Amsterdam, Paris and Venice image sets were used for search-by-image tasks. The Honolulu, Fiji and Stockholm image sets were used for search-by-text tasks. The Beijing, Seoul and Tokyo image sets were used for the brochure making task.

The order of the tasks was fixed because we didn't directly compare the usability for the tasks. To counterbalance order effect for the three views, we used the 3x3 Latin Square to decide the order, as shown in Table 1. The view index is 1,2,3 for overlapping, sliding and expanding, respectively.

Subject#		P1	P2	P3
Search by image	Amsterdam	1	2	3
	Paris	2	3	1
	Venice	3	1	2
Search by text	Honolulu	1	2	3
	Fiji	2	3	1
	Stockholm	3	1	2
Brochure making	Beijing	1	2	3
	Seoul	2	3	1
	Tokyo	3	1	2

Table 1. User Study Setup

Results and Discussion

In this section, we present results from the user study with both quantitative and qualitative data. We collected and analyzed the log data of image searching and browsing time, and the self-reported likert scale questions about the *efficiency*, *ease of use* and *enjoyment* for each of the three views.

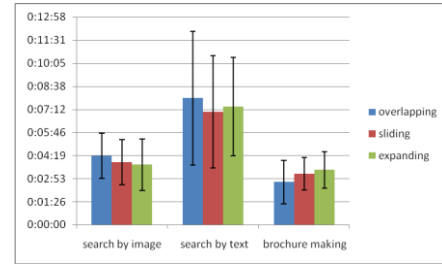


Figure 10. Average time of the three views

Search-by-image: As Figure 10 shows, the expanding view was slightly faster in the search by image task. We analyzed the logged data of search time by paired t-test and there were no significant difference between each pair of the three views.

By interviewing the participants, we found a common strategy that most of them applied. The users scanned the set of thumbnails by comparing certain features of the image, e.g. color, intensity and shape. Even if we made the pictures very small, most of them still managed to visually compare the image set with the targeting image without checking the details of the pictures one-by-one. As stated by P10, "I didn't use the interface much (for all three views), because I had the image in my head, I was looking for the mini version with them, I don't interact with them except looking for details; sometimes I look for colors, pattern searching." In this task, all the zooming techniques were not important, since the users tended to get around the interface. However, zoom-in-context views were still useful in the following situation: if there are several possible matched images that were close by, or if participants accidentally zoomed into a wrong image that was close to the one that they wanted to exam.

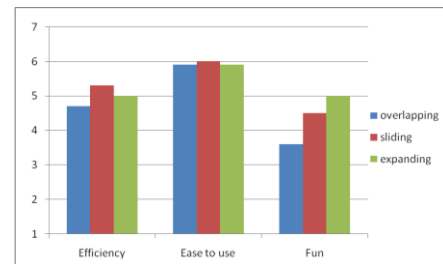


Figure 11. The average rating for the usability of three views in search-by-image task.

Search-by-text: As Figure 10 shows, the sliding view was slightly faster in the search by image task. We analyzed the logged data of search time by paired t-test and there were no significant differences between each pair of the three views.

With this task, the interface was used more often to find the image that fits the text-based description. The logged data showed a higher frequency of zooming into the pictures. The participants also stated that they realized that they need

to check the details more often than the in search-by-image task.

During the interview, we asked the participant to reflect about how they searched for the picture by text. They tended to “translate” the description to related objects, color and pattern and based their search on the translation. Here is one example: *“I have an image in mind... I had to find an anchor, I associated it with ships. Also, I don’t expect to see an anchor alone, so I am looking for water, which is blue.”* (P13)

Zoom-in-context views (sliding and expanding) were rated as more enjoyable and easier to use than non zoom-in-context views (overlapping) according to the paired t-test (enjoyment: expanding vs. overlapping, $t=2.31$, $p=0.01$. ease-of-use: sliding vs. overlapping, $t=1.83$, $p=0.05$).

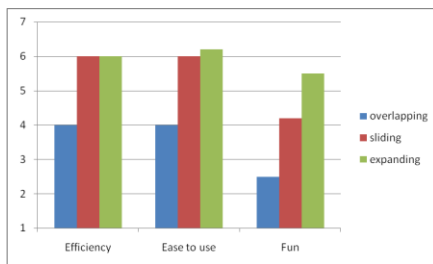


Figure 12. The average rating for the usability of three views in search-by-text task.

Brochure-making: In the brochure-making task, we encouraged the participants to spend as much time as they wanted while browsing and choosing the pictures. Users spent significantly more time using the expanding view than overlapping ($t=2.14$, $p<0.01$). This was related to the way users move from one picture to another. Several users said that they tended to check the pictures next to the expanded one. As stated by P12, *“Expanding view is the best for the brochure making. Because it’s non specific, it immediately draws my eyes to the area (around my mouse), I can just pick out interesting ones within that area.”* Another important reason was the match between the fluid feeling delivered by the animation of the expanding view and the nature of the brochure making task. P11 said, *“The expanding view feels more organic, while the two (views) are more rigid.”*; P2 said, *“the expanding view is more casual and has more fun, which is what the brochure making should be.”*

The data shows that in brochure making task, zoom-in-context views (sliding and expanding) were rated better for usability than non zoom-in-context view (overlapping) according to the paired t-test (efficiency: sliding vs. overlapping, $t=2.26$, $p=0.04$; ease-of-use: expanding vs. overlapping, $t=2.26$, $p=0.01$; enjoyment: expanding vs. overlapping, $t=2.31$, $p=0.03$). In overlapping view, the zoomed picture covers its neighbor pictures; the continuity of browsing adjacent images is interrupted.

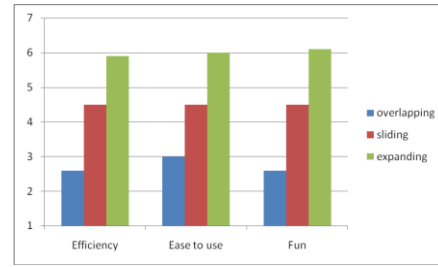


Figure 13. The average rating for the usability of three views in brochure task.

Discussion: In the user study, we explored three different tasks in two categories (searching and browsing). We found that people have different strategies and preferences of interface due to the different nature of the tasks. We found that (1) In search-by-image task, the three views had similar ratings in efficiency and ease of use. (2) The sliding and expanding views were rated more efficient and easier to use in search-by-text and brochure making. (3) The expanding view was the most enjoyable to use, especially in the brochure making task.

Below we explain these findings in the context of our observations from the study:

- *Structure and order are important for searching:* Both sliding and expanding view preserve neighbor relationship. However, participants preferred the sliding view over the expanding view in the two searching tasks. We attribute this preference to two factors: (1) sliding rigidly preserves the order; (2) people try to keep the order of image arrays in memory when they perform searching.
- *Zoom-in-context views are important when users really need to see details of images:* In search by text task, people need to select an image based on the text descriptions. They have to zoom in and see image details more frequently than in the search by image task. Similarly, for the brochure making, people have to zoom to see image details before choosing it. The zoom-in-context views are crucial since it keeps all neighbors visible in these tasks.
- *Expanding view is more preferred than overlapping view in brochure making.* We attribute this preference to two factors: (1) Zoom-in-context views are more suitable in a brochure making task; (2) expanding view delivers a unique fluid feeling of interaction, which is enjoyed by users in casual use.
- *The match between the interface and the tasks.* Our user study shows that, when developing image browsing interfaces, we need consider the nature of the task as an important element of the design.

CONCLUSION AND FUTURE WORK

We designed two zoom-in-context views, the sliding and expanding views for exploring large image collections. The two views enlarge a selected image while keeping all of its

neighbors visible. They also preserve the spatial relationship between the zoomed image and its neighbors. The sliding and expanding view are suitable for browsing and searching a large number of thumbnail images using a single screen. In addition, the expanding view creates a fluid feeling of interaction, which is preferred in casual image browsing. Our user study validates the ease of use of the two views.

In the future, we would like to continue to develop new zoom-in-context methods for images, perhaps including styles that make use of 3D arrangements and motion. We also want to incorporate our new zooming techniques in a more full-function image browsing system. In particular, we would like to see how effective these methods are in a system that allows a hierarchical organization of photos, and that also allows keyword search capabilities. Finally, it would be useful to perform user studies in which the subjects use our photo browsing techniques with their own photo collections over an extended period of time.

REFERENCES

1. Ames, M. and Manguy, L. "PhotoArcs: a tool for creating and sharing photo-narratives," *Ext. Abstracts CHI '06*, ACM Press (2006), 466-471.
2. Bederson, B. "PhotoMesa: A zoomable image browser using quantum treemaps and bubblemaps," *Proc. UIST'01*, ACM Press (2001), 71-80.
3. Cooper, M., Foote, J., Girgensohn, A., and Wilcox, L. "Temporal event clustering for digital photo collections," *Proc. ACM Multimedia '03*, ACM (2003), 364-373.
4. Furnas, G. W., Generalized fisheye views. In *Proc. CHI '86*, ACM Press (1986), 16-23.
5. Graham, A., H. Garcia-Molina, A. Paepcke, and T. Winograd. Time as essence for photo browsing through personal digital libraries. In *Proc. Joint Conf. Digital Libraries*, 2002.
6. Hausner, A. "Simulating Decorative Mosaics," *Proc. SIGGRAPH 2001*, 573-580.
7. Hoff, K., Keyser, J., Lin, M., Manocha, D. Culver, T. "Fast Computation of Generalized Voronoi Diagrams Using Graphics Hardware," *Proc. SIGGRAPH 99*.
8. Huynh, D., Drucker, S., Baudisch, P., and Wong, C. "Time quilt: Scaling up zoomable photo browsers for large, unstructured photo collections," *Ext. Abstracts CHI'05*, ACM Press (2005), 1937-1940.
9. Kang, H.; Shneiderman, B., Visualization methods for personal photo collections: browsing and searching in the PhotoFinder. In *IEEE Int'l Conf. Multimedia and Expo, 2000*. Vol3, 1539-1542.
10. Kustanowitz, J. and Shneiderman, B. Meaningful presentations of photo libraries: Rationale and applications of bi-level radial quantum layouts. In *Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries* (2005). ACM Press, NY, 188-196.
11. Mackinlay, Jock D., George G. Robertson, Stuart K. Card, The perspective wall: detail and context smoothly integrated, In *Proc. CHI'91*, 173-176.
12. Platt, J., Czerwinski, M., and Field, B. "PhotoTOC: Automatic clustering for browsing personal Photographs," *Microsoft Research Tech. Rep. MSR-TR-2002-17* (2002).
13. Robertson, G. G., Mackinlay J.D, Card, S.K., Cone Trees: animated 3D visualizations of hierarchical information, In *Proc. CHI'91*, 189-194.
14. Robertson, G. G., Mackinlay, J. D. The Document Lens. In *Proc. UIST'93*, 101-108.
15. Rodden, K., Basalaj, W., Sinclair, D. and Wood, K., "Does Organisation by Similarity Assist Image Browsing?" In *Proc. CHI 2001*, 190-197.
16. Rodden, K. and Wood, K. "How do people manage their digital photographs?" In *Proc. CHI 2003*, 409-416.
17. Sarkar, M., Brown, M. Graphical fisheye views of graphs, In *Proc. CHI 1992*. 83-91.
18. Shneiderman, B., Bederson, B., and Drucker, S. Find that photo! Interface Strategies to Annotate, Browse, and Share. *Communication of ACM*, 49(4), 2006, 69-71.
19. Smith, G., Czerwinski, M., Meyers, B, Robbins, D., Robertson, G., Tan, D. "FacetMap: A Scalable Search and Browse Visualization," *IEEE TVCG*, vol. 12, no. 5, 797-804, 2006.
20. Torres, R., Silva, C., Medeiros, C., and Rocha, H. "Visual structures for image browsing," *Proc. CIKM'03*, ACM Press (2003), 49-55.
21. Yang, D., Rundensteiner, E.A., Ward, M.O. "Analysis Guided Visual Exploration of Multivariate Data". *VAST 2007*, 83-90.